中文“非多字词错误”自动校对方法研究

刘亮亮[1] 曹存根[2]

(江苏科技大学计算机科学与工程学院 镇江 212003)[1]

(中国科学院计算技术研究所智能信息重点实验室 北京 100190)[2]

摘 要 针对目前中文文本中的“非多字词错误”自动校对方法的不足,提出了一种模糊分词的“非多字词错误”自动查错和自动校对方法。首先利用精确匹配算法与中文串模糊相似度算法对中文文本进行精确切分和模糊全切分,建立词图;然后利用改进的语言模型对词图进行最短路径求解,得到分词结果,实现“非多字词错误”的自动发现和自动纠正。实验测试集是由2万行领域问答系统日志语料构成,共包含664处“非多字词错误”。实验表明,所提方法能有效发现“非多字词错误”,包括由于汉字替换、缺字、多字引起的“非多字词错误”,该方法的查错召回率达到75.9%,查错精度达到85%。所提方法是一种将查错与纠错融于一体的方法。

关键词 非词错误,非多字词错误,模糊匹配,词图

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.10.038

Study of Automatic Proofreading Method for Non-multi-character Word Error in Chinese Text

LIU Liang-liang[1] CAO Cun-gen[2]

(School of Computer Science & Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China)[1]

(Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)[2]

Abstract Aiming at the insufficiency existing in current automatic proofreading method of Chinese non-multi-character word error, a method based on fuzzy segmentation of automatic error-detecting and automatic proofreading for 'non-multi-character word error' in Chinese texts was proposed. Firstly, we used exact matching algorithm for exact word segmentation and fuzzy matching algorithm for Chinese strings to full fuzzy segmentation, and built the word graph. In order to get the best segmentation results, a method based on an improved language model is used to solve the shortest path. After that, the 'non-multi-character word errors' in Chinese texts are achieved to be automatic error-detected and automatic proofreaded. The test set has 20000 sentences of domain question answering log which include 664 non-multi-character word errors. Experiments show that the method proposed in our paper can detect the 'non-multi-character word error' effectively including character substitution errors, deletion errors and insertion errors. The error-detection recall rate is 75.9% and the error-detection accuracy rate is 85%. The proposed method combines automatic error-detecting and automatic error-correction together.

Keywords Non-word error, Non-multi-character word error, Fuzzy matching, Word graph

1 引言

随着信息处理技术与互联网技术的高速发展,传统的工作几乎全部被计算机所取代,电子书、电子报纸、电子邮件、办公文件、微博、博客等成为人们生活中的一部分,然而文本中的错误越来越多,这给校对工作带来了很大的挑战。文字错误是文本错误中最主要的一种。早在20世纪60年代,国外就开展了英文文本自动校对研究[1],目前已有商用的系统(例如Word中的拼写和语法功能)。

英文文本自动校对是以词为中心的,分为两种类型,一种是非词错误,一种是真词错误[1]。不在词典中出现的词称为非词错误。

例1 The bok is on the desk.

例1中“bok”不是词典中的词,应该为“book”,“bok”为非词错误。

在词典中出现,但是不适用当前的上下文的词的错误,称为真词错误,也称为上下文相关错误。

例2 This chocolate cake is a famous desert.

例2中的“desert”就是真词错误,应为“dessert”,desert也是词典中的词。

中文文本不像英文一样有空格对词进行隔开,并且输入到计算机中的中文文本的汉字本身不会发生错误,因为汉字是通过输入法输入到计算机的,因此中文文本不会像英文一样出现“非词错误”。但是根据汉语多字词发生错误后,经过

到稿日期:2015-09-14 返修日期:2015-12-07 本文受国家自然科学基金项目(91224006,61173063,61203284,30973713),国家社科基金重点项目(10AYY003)资助。

刘亮亮(1979—),男,博士,讲师,主要研究领域为自然语言理解、知识工程与知识获取,E-mail:lingyun79626@126.com;曹存根(1964—),研究员,主要研究领域为知识工程。

分词工具分词后的特点，张仰森等提出中文文本中的“非多字词错误”和“真多字词错误”[2]。“非多字词错误”指一个中文多字词(汉字个数大于或等于2)由于一个或多个汉字出现别字替换错误、缺字错误或多字错误后不是词典中的词，其最大的特征是在分词的过程中会分成长度大于或等于2的散串。例如，“无缘无顾”就是一个“非多字词错误”，在分词的过程中会分成散串“无缘”、“无”、“顾”。而中文文本中的“真多字词错误”指一个多字词错成词典中的另一个多字词，例如，“接收总统的邀请”中的多字词“接收”是一个“真多字词错误”，其正确的词是“接受”。

中文多字词一般会出现字替换错误(也称为别字错误)、缺字错误(丢失一个或多个汉字)、多字错误(插入了一个或多个汉字)，因此“非多字词错误”又包括“别字型非多字词错误”、“缺字型非多字词错误”以及“多字型非多字词错误”。本文将这3种类型统称为“非多字词错误”。而本文提出的方法主要是自动发现和自动校对“非多字词错误”。

目前的中文文本自动校对研究主要利用二元或三元语法对分词后的散串发现文本中的错别字，由于数据稀疏等原因，其召回率和准确率都不高，并且现有的方法只是局限于查错，无法给出修改建议。由于“非多字词错误”在分词过程中会分成多个散串，破坏了词本身的结构，并且中文本身存在单字词，因此需要用高阶的ngram模型去判断散串是否是合理的散串，使用高阶的ngram模型会带来更严重的数据稀疏。并且，对于长词错误，如果割裂开来进行判断，一方面会导致漏判，另一方面会对错误的位置提示不准确，从而难以给出合理的修改意见，即无法实现自动校对。

2 相关工作

对于英文的“非词错误”，一般都是采用ngram分析方法和查字典等方法进行查错和校对，而中文的“非多字词错误”的查错方法一般采用统计的方法来进行查错。

啄木鸟系统[3]是国内出现得比较早的中文自动校对系统。该系统的自动查错方法的出发点是文本中的绝大多数错误都导致切分后出现单字词，根据单字词的词频和它与前后两个汉字的接续强度给该单字词打分，最后将得分与一个预先设定的阈值进行比较来判断该单字词是否为错字。该系统的召回率在70%左右，准确率只有2.5%。张照煌等根据4种汉字的相似类型对每个句子中的每个汉字用其相似子集中的汉字依次替换，然后用词间字二元模型和词性的二元模型对各字串进行评分，最后选出得分最高的字串[4]。该方法是查错和纠错一体的方法，召回率达到76.64%，准确率为51.72%。

张磊等[5]提出基于特征和Winnow学习模型的中文自动校对方法，首先定义字或词的混淆集，然后提取目标串的二元接续关系、词性类的三元接续关系、上下文语义类、词性类邻接字等4种特征集，根据Winnow方法进行特征学习，然后利用这些上下文特征对目标词混淆集中的词进行选择。实验表明，该方法召回率达到85%，准确率达到41%，校对准确率达到51%，性能比目前常用的词的N元模型方法有明显的提高，该方法是查错和纠错一体的方法。

马金山等[6]构建了一种多方法融合的中文自动校对模型，该模型以三元模型为基础，对文本进行局部分析，以查找文本中的局部错误。同时利用依存文法的特点，对句子进行依存分析，在查找全局错误的研究上提出了新的思路和方法；张仰森于2006年提出一种基于规则和统计相结合的方法[7]，根据正确文本分词后单字词的出现规律，提出一组错误发现规则，并与针对分词后单字散串建立的字二元、三元统计模型和词性二元、三元统计模型相结合，建立了文本自动查错模型与实现算法。与张仰森[8]提出的基于接续关系的字词查错模型相比，基于规则与统计相结合的查错模型的查错准确率更好。该方法主要从分词后的单字散串入手，对于一些单字词错误和字替换后不成词的错误(“非词错误”)等有很好的效果，但对于“计算机拥护”之类的“真词错误”无法判断。该方法的查错召回率为86.85%，查准率为69.43%，误报率为30.57%。

目前的方法对“非多字词错误”都是根据分开后的散串利用二元或三元以及上下文特征来进行判断。张磊等提出一种快速的中文模糊词匹配的方法，利用相似字替换和语言模型评分的自动校对方法，改进了张照煌的基于相似字集的自动校对方法，能发现多字、缺字和字串替换等错误[9]。刘亮亮等提出一种基于散串合并与统计验证的方法来发现中文文本的错别字，该方法对领域问答系统日志进行分词，对分词中的多字词和合并的串进行相似词串聚类，对相似词串的上下文语境进行统计分析，从中自动获取错别字对，该系统获得了71.32%的召回率和82.6%的准确率[10]。

3 本文的方法

词是最小的能够独立活动的有意义的语言成分[11]。因为中文中词与词没有类似英文明显的区分标记，并且中文中存在单字词和多字词，因此中文词语分析是中文信息处理的基础与关键[12]。中文词语的分析首要是中文分词。文本校对系统中的分词是为文本校对服务的，因此中文文本校对系统的分词应具备一定的查错和纠错能力。如果一个多字词中存在错别字，在分词的过程中会出现单字散串，但是出现单字散串不代表这个散串中一定会出现错别字，原因之一是汉字单字成词的能力非常强，很多汉字都可以单字成词，例如，“我”、“的”等都是单字词，另外一个原因是中文未登录词一般都会分成散串。如果仅仅对散串采用模糊匹配的方法，误判率会非常高。

例3 我的电脑无缘无故的死机了？

分词结果：我的电脑无缘无故的死机了？

模糊匹配：我的电脑无缘无故的司机了？

分析：“死机”是一个单字散串，如果利用模糊匹配，则会匹配上“司机”，但是“死机”是一个合理的单字散串。

例4 用飞信发短信会收费吗？

分词结果：用飞信发短信会收费吗？

模糊匹配：用飞行发短信会收费吗？

分析：“飞信”是一个未登录词，但是经过模糊匹配后，会得到词“飞行”。

3.1 基本思想

首先利用最大匹配方法对文本进行粗切分；然后对切分后的文本中的散串做如下分析：利用二元和三元语言模型对切分后的散串进行判断，如果切分后的散串的二元和三元大于一定的阈值，那么认为该散串是合理的散串；否则利用模糊匹配算法对散串进行模糊匹配，利用中文词串的相似度公式计算相似度，从而得到其相似的模糊词，将该词加入到切分结果中，建立模糊词图。利用语言模型求模糊词图的最短路径，

从而得到分词结果。如果分词后的串和原文本一致，则认为原文本没有错误，如果存在模糊匹配的词，则发现“非多字词错误”，其模糊匹配的词即为修改建议。

假设待分词的句子 $S=c_1c_2\cdots c_n$，其中 $c_i(i=1,2,\cdots,n)$为单个字，n 为句子的长度，$n>1$，对 S 进行最大匹配粗切分后，得到 $S=W_1W_2\cdots W_m$，采用文献[12]中的方法建立词图[12]，在建立词图的过程中，对散串进行分析，采用中文词串的相似度算法(式 3)获得散串的相似词。建立模糊词图的方法和步骤如下。

建立节点数为 $m+1$ 的切分有向无环图 G，各节点编号依次为 $V_0,V_1,V_2,\cdots,V_m$。依次扫描粗切分的结果，通过以下方法来建立该图上所有的边：

(1)相邻节点 V_i 和 V_{i+1}之间建立有向边$\langle V_i,V_{i+1}\rangle$；

(2)如果 V_i 到 V_j 通过模糊匹配算法可以模糊成词，那么节点 V_i,V_j 之间建立有向边$\langle V_i,V_j\rangle$；

对 S 粗切分的结果进行扫描和模糊匹配后，得到一个有向无环图。

如图 1 所示，首先对句子进行向后最大匹配切分，然后利用规则和统计数据对句子中的散串进行模糊匹配，图中的实线框表示精确匹配的分词，虚线框为调用模糊匹配的结果，最后粗线条的路径即为最优路径。

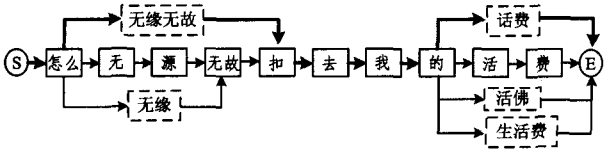

图 1　词图示例

3.2　中文串的相似度计算

中文串由汉字构成，因此中文词串的相似度可以通过汉字间的相似度来定义，而汉字的相似可以分为音相似度和形相似度。因此，对于两个汉字 c_i 和 d_i，其相似度定义如下：

$$sim(c_i,d_i)=\begin{cases}1, & \text{if } c_i=d_i\\ 0, & \text{if } c_i=\varepsilon \text{ or } d_i=\varepsilon,\varepsilon \text{ 是空字符}\\ \alpha*PSim(c_i,d_i)+\beta*SSim(c_i,d_i), & \text{其它}\end{cases}\quad(1)$$

其中，$PSim(c_i,d_i)$表示汉字 c_i 和 d_i 的拼音相似度[13]，$SSim(c_i,d_i)$表示汉字 c_i 和 d_i 的形相似度[14]。参数 α 和 β 分别表示拼音相似度和形相似度的权重，满足 $\alpha+\beta=1$，其可以根据具体的应用来设定。例如，如果是拼音输入法，可以设 $\alpha=1$，$\beta=0$；如果是 OCR 识别后纠错，可以设定 $\alpha=0,\beta=1$。

根据字符串编辑距离的定义，中文串的编辑操作包括多字、缺字及汉字替换，根据前面的分析，汉字的编辑距离通过如下方法计算：

$$editdis(c,d)=\begin{cases}1, & \text{if } c \text{ 为空或 } d \text{ 为空}\\ 0, & \text{if } c=d\\ 1-sim(c,d), & \text{否则}\end{cases}$$

因此，两个中文串 $W_1=c_1c_2\cdots c_n,W_2=d_1d_2\cdots d_m$ 的距离函数为：

$$editdis(W_1,W_2)=\max\begin{cases}editdis(c_2\cdots c_n,d_1\cdots d_m)+1\\ editdis(c_1\cdots c_n,d_2\cdots d_m)+1\\ editdis(c_2\cdots c_n,d_2\cdots d_m)+(1-sim(c_1,d_1))\end{cases}\quad(2)$$

从而，编辑距离越大，相似度越低，因此我们定义两个中文串的编辑距离相似度 $Sim(W_1,W_2)$为：

$$Sim(W_1,W_2)=1-\frac{editdis(W_1,W_2)}{\max(m,n)}\quad(3)$$

3.3　精确匹配算法

中文文本自动校对需要大量的知识与词典，其中有一类知识是错字词词典。错字词词典是由错误词串及其对应的正确的词来构成的。其定义如下。

定义 1(错字词)　可以定义为元组 $WP=(WrongPhrase,RightPhrase)$，其中 $WrongPhrase$ 为一包含错别字的串，$RightPrase$ 表示 $WrongPhrase$ 对应正确的词。

例如，$WP_1=$(万花同，万花筒)。错字词词典可以通过人工收集常用的错词和通过自动获取来构建。刘亮亮等提出基于散串合并的错误发现方法，该方法能最终生成错字词知识[10]。本文实验中，为了验证所提方法的有效性，将错字词词典置为空。

因此，在分词过程中参与分词的词典有正确词词典与错字词词典。在粗切分精确分词过程中，需要同时考虑两个词典的匹配。在前向最大匹配的分词上，做如下改进：先精确匹配正确词词典，记录结果；然后利用错字词词典进行匹配。

例如：$S=$“…无原无故…”，假如正确词词典＝{无，原，无故，无缘无故}，错字词对词典＝(〈无原无故，无缘无故〉)。

分词过程：首先利用正确词词典进行分词，得到“无”，同时利用错字词词典进行匹配，得到“无原无故”，通过错字词对找到其正确的词“无缘无故”，则把“无缘无故”加入到词图中，并对错字词词典中匹配的词进行标记，然后从“原”开始进行类似的操作。最后得到的分词结果是“无，原，无故，无缘无故”。重复这样的过程建立词图。利用错字词词典进行匹配也认为是精确匹配。本文词典算法采用双数组 Trie 树结构来建立正确词词典和错字词词典[15]。

精确匹配算法如算法 1 所示。

算法 1　Accurate-Matching

输入：S 为待切分句子；Dic1 为正确词典，Dic2 为错字词对词典；Pos1 为正确词典查找位置；pos2 为错字词词典查找位置

输出：词图

1. Begin
2. pos1←0
3. pos2←0
4. 利用正确词典 Dic1 从 pos1 位置前向最大搜索，假设搜索出词条 word1，将其加入词图，pos1 更新为 word1 之后的位置
5. 利用错误词典 Dic2 从 pos2 位置前向最大搜索，若搜索出词条 word2，将其对应的正确词条 word 加入词图，pos2 更新为 word2 之后的位置；否则 pos2 指向当前位置的下一个字
6. Repeat 3、4 Until，pos1 与 pos2 均指向 sent 句尾
7. End

3.4　模糊匹配算法

中文词串的模糊匹配算法是给定句子和起始匹配的位置，在词表中查找所有与该位置处开始的各种长度的子串的相似度不小于阈值 t_w 的候选词，并且给出两个词串之间的相似度和下一次匹配的位置。

算法从当前位置 nCurr 开始，读入当前字符，对当前字符进行模糊匹配，同时计算两个子串的相似度，两个子串的相似度采用式(2)来计算，如果相似度不小于阈值 t_w，那么记录下一个要读入的位置，在模糊的过程中，当前位置的字可以是单字词替换，也可以是多字或缺字来计算相似度。从算法中可以看到，在当前位置模糊匹配时，算法对当前位置的汉字进行

替换、加字和减字来进行模糊匹配，最后得到一组相似度的词和相似度（*sFuzzyWord*，*next*，*sim*），其中，*sFuzzyWord* 为匹配上的词，*next* 为下一个的位置，*sim* 是相似度。由于利用相似度阈值 t_w，大大限制了搜索空间的大小。

利用模糊匹配算法能快速地匹配类似于“无缘无辜（正确词：无缘无故）”这种“别字型非多字词错误”，同时也可以匹配类似于“一生平安全（正确词：一生平安）”这种“多字型非多字词错误”，以及“莫名妙（正确词：莫名其妙）”这种“缺字型非多字词错误”。具体算法如算法 2 所示。

算法 2 FuzzyMatch(sent, nCurr, sFuzzyWord, sim, result)

输入：sent 为待分词句子；nCurr 为开始匹配位置；sim 为相似度阈值

输出：result 为从当前位置 nCurr 所有模糊匹配的词

```
1. Begin
2. sFuzzyWord="",sim=1,result=NULL;//sFuzzyWord 为模糊词
3.    if sim<threshold then //如果相似度小于一定阈值直接返回
4.      return
5.    end if
6.    if sFuzzyWord 是词典中的词条 then
7.      将模糊词加入到模糊列表中;
8.    end if
9.    if 当前位置为句子末尾 then
10.    if sFuzzyWord 后可以再接字 then
11.      for ch in sFuzzyWord 后可以接的字集合 do
12.        FuzzyMatch(sent,nCurr,sFuzzyWord+word,sim * α,
           result) //尝试增加一个字
13.      end for
14.    end if
15.    return
16.  end if
17.  if sFuzzyWord 后可以接当前句子字 sent(nCurr) then
18.      FuzzyMatch(sent,nCurr+1,sFuzzyWord+word,sim,re-
         sult)//精确匹配
19.  end if
20.  FuzzyMatch(sent,nCurr+1,sFuzzyWord,sim * α,result)//跳
     过当前字，是否为多字错误
21.  for ch in sFuzzyWord 后可以接的字集合 do
22.    if IsSimilar(word,sent(nCurr)) && sim * α<threshold then
23.      FuzzyMatch(sent,nCurr+1,sFuzzyWord+word,sim *
         α,result);
24.    end if
25.    FuzzyMatch(sent,nCurr,sFuzzyWord,sim * α,result)//当前
       位置加一个字，尝试缺字错误
26.  end for
27. End
```

例如，S=“怎么无原无故扣除我的费用”，当前位置从“无”开始进行模糊匹配，通过混淆集找到可能写错成“原”的混淆字“元，缘…”，然后替换“原”，从双数组 Trie 树中查找，发现“无缘”匹配成功，但是双数组 Trie 树中的“无缘”不是终结状态，并且“无缘”与“无原”的相似度在阈值 t_w 范围内，因此继续读入匹配，重复以上过程，可以匹配上“无缘无故”。同时尝试在当前汉字处添加一个字和删除“原”字来进行模糊，发现没有模糊匹配成功的词，因此最后的模糊结果为（无原无故，无缘无故），同时下一次模糊的位置为“原”，重复以上的模糊过程，找到句子中所有的模糊匹配的词。

通过模糊结果，可以利用训练语料统计信息对模糊的结果进行后处理，为了提高模糊分词纠错的准确率，本文对模糊分词结果采用如下方法进行后处理。

1）如果模糊的串是一个高频串，那么放弃该模糊的结果，这是为了避免一些合理的散串被模糊成词，反而出错。

2）如果模糊的串中包含了多个高频的单字成词的字，那么放弃该模糊匹配的结果。

根据精确匹配算法与模糊匹配算法对句子进行模糊全切分，如算法 3 所示。

算法 3 Fuzzy-All-Segement

输入：待切分句子 sent，正确词典 Dic1，错字词对词典 Dic2

输出：模糊全切分词图

```
1. Begin
2.   pos=0
3.   n=sent.length
4.   for i=0 to n do
5. 调用 FuzzyMatch 查找 i 位置开始的所有 Dic1 中的词，加入到词图中
6.     if pos==i then
7.       if 在 pos 处前向最大搜索匹配到 Dic2 中的词 word then
8. 将 word 在 Dic2 中对应的正确词条 word2 加入词图，pos 更新为
   word 之后的位置
9.         else
10.          pos++
11.      end if
12.    end if
13.  end for
14. End
```

3.5 最短路径求解

对句子 S 进行精确分词与模糊切分后，得到多条路径，需要用语言模型对每条路径进行计算，得到每种切分后序列的概率，然后选择概率最大的路径，即为最后的切分结果。即对每一种切分结果，采用语言模型计算其概率，并输出概率最大的切分结果。本文采用词的二元语言模型来计算切分后的概率。

$$\begin{aligned} W^* &= \arg\max_W P(W) \\ &= \arg\max_W P(W_1 W_2 \cdots W_n) \\ &= \arg\max_W p(W_1)\prod_{i=2}^{n} p(W_i \mid W_{i-1}) \end{aligned} \tag{4}$$

由于采用了模糊切分，因此不能直接用语言模型进行计算，需要对模糊切分加上一定的惩罚，在词图中对模糊切分的词的转移概率乘以一个惩罚值 α，该惩罚值 α 可以取为模糊的词与串的相似度的值。加上惩罚值后，式(4)改写成式(5)：

$$\begin{aligned} W^* &= \arg\max_W P(W) \\ &= \arg\max_W p(W_1)\prod_{i=2}^{n}(p(W_i \mid W_{i-1}) * \alpha(W_{i-1}, W')) \end{aligned} \tag{5}$$

如果当前词是精确切分，$\alpha(W_{i-1}, W')=1$；否则 $\alpha(W_{i-1}, W')=sim(W_{i-1}, W')$，即模糊匹配的串 W' 与匹配上的词 W_{i-1} 的相似度。

为了避免分词过程中的过纠错，需要限定句子的错误程度，除了可以通过模糊匹配的相似度限定错误程度以外，还可以设定句子可能的错误个数。对语料中的错误句子进行分析，一个句子中错别字的个数大部分都只是一个或两个，因此，根据句子的长度，设定如下假设：句子的错误个数不超过句子内汉字个数的 20%，也就是每一条路径中模糊匹配的个数不超过汉字个数的 20%。最后利用图的 Dijkstra 算法来求

解模糊词图中的最短路径。

在求得的分词结果中，通过模糊匹配得到的结点即表示该结点对应的原串是一个非多字词错误，而模糊匹配上的词即为其可能正确的词，从而实现了非多字词错误的自动发现与自动校对。

4 实验结果与分析

由于目前错别字识别缺乏标准测试集，而大多数研究的测试集是人为构造错误的测试集，人为构造错误的测试集有很大的偏向性，不能反映错误的真实情况。领域问答系统的用户咨询中包含大量的文本错误，有大量的“别字型非多字词错误”、“多字型非多字词错误”以及“缺字型非多字词错误”。因此，利用某领域问答系统的用户咨询日志并通过人工标注来构造测试集。本文的测试集随机抽取 2 万行咨询日志，人工对咨询日志中的“非多字词错误”进行标注，形成测试集。该测试集中包含 664 处“非多字词错误”，其中“非多字词错误”包括“别字型非多字词错误”、“多字型非多字词错误”以及“缺字型非多字词错误”。错误分布和实验结果如表 1 所列。

表 1 错误分布与实验结果

	二字词替换错误	三字词以上替换错误	三字词以上插入错误	三字词以上删除错误	总计
文本错误	415	204	35	20	664
本方法	325/172	202/200	32/28	16/12	504/412

注：正确的查错数/正确纠正的错误数。

错别字识别有两个过程，一是对文本中的错别字进行识别，因此在错别字识别过程中主要依据如下性能指标：错别字识别的查错召回率（*Recall Rate*）、查错精度（*PrecisionRate*），以及查错的 *F* 度量（*F-Score*），定义如下。

$$Recall=\frac{\text{正确发现错误的总数}}{\text{文本中的错误总数}}*100\%$$

$$Precision=\frac{\text{正确发现错误的总数}}{\text{发现错误的总数}}*100\%$$

$$FScore=\frac{2*Recall*Precision}{Recall+Precision}*100\%$$

另一个阶段是对识别的错误进行校对，给出修改建议，主要依据纠错的准确率（Correct Accuracy Rate）和纠正率（Correct Rate）两个指标来评价校对的性能。在纠错过程中，如果修改意见中有正确的词，则认为正确纠错。

$$Accuracy=\frac{\text{正确纠错的总数}}{\text{正确识别的总数}}*100\%$$

$$Correct\_rate=Accuracy*Reca$$

$$=\frac{\text{正确纠错的总数}}{\text{文本中的错误总数}}*100\%$$

根据以上两个阶段的评价指标的计算方法，得到的实验结果的评价指标值如表 2 所列。

表 2 评价指标结果

查错总数	查错召回率	查错精度	纠正率	纠错准确率	查错 F 值
593	75.9%	85%	62%	81.7%	80.2%

根据表 1 和表 2 的实验结果来看，基于模糊分词的错别字识别的精度非常高，改错的准确率也非常高，在正确识别的错别字中，81.7%的错别字被正确改正，并且该方法对长词错误的识别和校对非常有效，这是本文方法的一大优点和目标。

表 3 部分实验结果

例句 1：昨天为什么无源无故扣了两元？
精确切分：昨天为什么无源无故扣了两元？
模糊切分：昨天为什么无缘无故扣了两元？
例句 2：怎样拔打异地电话比较优惠？
精确切分：怎样拨打异地电话比较优惠？
模糊切分：怎样拨打异地电话比较优惠？
例句 3：我要去消超级 QQ 业务怎么去消
精确切分：我要去消超级 QQ 业务怎么去消
模糊切分：我要取消超级 QQ 业务怎么取消
例句 4：我这么忠耿耿的用你们移动公司的号。
精确切分：我这么忠耿耿的用你们移动公司的号。
模糊切分：我这么忠心耿耿的用你们移动公司的号。
例句 5：怎么扣了我那么多的话说费？
精确切分：怎么扣了我那么多的话说费？
模糊切分：怎么扣了我那么多的话费？
…

自动校对的两类问题：漏报与误报。根据实验分析，本方法的漏报主要有两个方面：

（1）出现的错别字的错误程度太大，也就是说错别字与其对应正确的相似的字的相似度太低。例如，句子“你帮我查一下是不是我的 GPRS 超油量?”的“油”就是一个错别字，其正确的字为“流”（注：“油量”不是词典中的词，“流量”是字典中的词），而“油”与“流”相似程度低。这在模糊匹配时无法匹配上。对于这类错误，只能通过上下文来判断，并且对错误进行标记，很难进行自动校对。

（2）模型的数据稀疏。发生错误的词分词后成散串，而散串与前后词的接续强度大，而正确的词与前后词关系接续强度弱，从而会导致漏判。例如，“科谱端口怎么使用?”统计模型中“科普端口”共现为 0，而“谱端口”、“科谱”共现强度较大，从而无法对该类错误进行召回。

模型的数据稀疏也是误报的一个主要原因。例如，句子“对方的短号想同我联网，还要收费吗?”中，本方法会将散串“想同”模糊成“相同”，并且“短号相同”、“相同我”接续关系都很大，从而导致误判。

根据实验结果可以看到，基于模糊分词的错别字识别方法有效地解决了长词（字数大于或等于 3）的替换错误、多字错误及删除错误，对于二字词的查错能力有限，会带来一定的误判，因此本文的模糊分词的错别字识别方法主要是解决汉语中的长词错误问题，为了提高对二字词的查错准确率，可以对二字词的模糊匹配提高其相似度的阈值以及加强上下文验证来控制二字词的模糊匹配。

结束语 由于汉字必须通过输入法输入到计算机中，因此中文文本不像英文文本一样会出现“非词错误”。本文首先根据中文词发生错误后的特点，将中文多字词错误分为“非多字词错误”与“真多字词错误”。然后提出一种基于模糊分词的中文“非多字词错误”的自动校对方法，该方法对中文文本进行精确切分，同时利用中文串的模糊匹配算法对句子进行模糊切分，将精确切分的词和模糊切分的词建立词图，然后利用改进的语言模型对模型进行求解，求得最短路径，从而实现“非多字词错误”的自动发现与自动校对。实验结果表明，本文的方法能有效地发现中文文本中的“非多字词错误”，查错召回率达到 75.9%，查错精度达到 85%。针对实验中的问题，下一步将从以下几个方面进行“非多字词错误”自动校对研究：1）对模糊匹配的结果加强上下文语境的验证，除了利用直接的上下文语境以外，还需要用到搭配等语境来进行进一步的验证；2）利用当前查错文档的统计信息来进行判断和验证。当前查错文档中具有丰富的统计信息，能减少数据稀疏

带来的问题。特别是未登录词在当前查错文档中会出现多次，从而减少由于未登录词而带来的误判。

参 考 文 献

[1] Kukich K. Techniques for automatically correcting words in text [J]. ACM Computing Surveys (CSUR), 1992, 24(4): 377-439

[2] Zhang Yang-sen, Cao Yuan-da, Yu Shi-wen. A Hybrid Model of Combining Rule-based and Statistics-based Approaches for Automatic Detecting Errors in Chinese Text[J]. Journal of Chinese Information Processing, 2006, 20: 1-7(in Chinese)
张仰森，曹元大，俞士汶. 基于规则与统计相结合的中文文本自动查错模型与算法[J]. 中文信息学报，2006，20：1-7

[3] Shi De-sheng, Wang Liang-zhi, Chen Zhi-da, et al. A Statistics-based Approach for Automatic Detecting Errors in Chinese Text [J]. Computer and Communications, 1992, 8: 19-26(in Chinese)
施得胜，王良志，陈志达，等. 基于统计的中文错字侦测法[J]. 电脑与通讯，1992，8：19-26

[4] Zhang Zhao-huang. Automatic Error Detection and Correction of Chinese Text[J]. Communications of COLIPS, 1994, 4(2): 143-149(in Chinese)
张照煌. 中文错别字自动订正方法初探[J]. Communications of COLIPS, 1994, 4(2): 143-149

[5] Zhang L, Zhou M, Huang C, et al. Multifeature-based approach to automatic error detection and correction of Chinese text[C]// Proceedings of the First Workshop on Natural Language Processing and Neural Networks. 2000: 2744-2748

[6] Ma Jin-shan, Zhang Yu, Liu Ting, et al. Detecting Chinese Text Errors Based on Trigram and Dependency Parsing[J]. Journal of the China Society for Scientific and Technical Information, 2005, 23(6): 723-728(in Chinese)
马金山，张宇，刘挺，等. 利用三元模型及依存分析查找中文文本错误[J]. 情报学报，2005，23(6)：723-728

[7] Zhang Yang-sen, Cao Yuan-da, Yu Shi-wen. A Hybrid Model of Combining Rule-based and Statistics-based Approaches for Automatic Detecting Errors in Chinese Text[J]. Journal of Chinese Information Processing, 2006, 20(4): 1-7(in Chinese)
张仰森，曹元大，俞士汶. 基于规则与统计相结合的中文文本自动查错模型与算法[J]. 中文信息学报，2006，20(4)：1-7

[8] Zhang Yang-sen, Ding Bing-qing. Automatic Errors Detecting of Chinese Texts Based on the Bi-neighborship[J]. Journal of Chinese Information Processing, 2001, 15(3): 36-43(in Chinese)
张仰森，丁冰青. 基于二元接续关系检查的字词级自动查错方法[J]. 中文信息学报，2001，15(3)：36-43

[9] Zhang Lei, Sun Mao-song, Zhou Ming, et al. Automatic Chinese Text Error Correction Approach Based on Fast Approximate Chinese Word-Matching Algorithm[C]// Procceding of the 3rd World Congress on Intelligent Control and Automation. 2000: 2739-2743(in Chinese)
张磊，孙茂松，周明，等. 基于快速模糊词匹配算法的中文自动校对方法[C]//第三届全球智能控制与自动化大会. 2000：2739-2743

[10] Liu Liang-liang, Wang Shi, Wang Dong-sheng, et al. Automatic Text Error Detection in Domain Question Answering[J]. Journal of Chinese Information Processing, 2013, 27(3): 77-83(in Chinese)
刘亮亮，王石，王东升，等. 领域问答系统中的文本错误自动发现方法[J]. 中文信息学报，2013，27(3)：77-83

[11] 朱德熙. 语法讲义[M]. 商务印书馆，1982

[12] Zhang Hua-ping, Liu Qun. Model of Chinese Words Rough Segmentation Based on N-Shortest-Paths Method[J]. Journal of Chinese Information Processing, 2002, 16(5): 1-7(in Chinese)
张华平，刘群. 基于 N-最短路径方法的中文词语粗分模型[J]. 中文信息学报，2002，16(5)：1-7

[13] 丰强泽，曹存根. 语音查询中的辨音方法[M]. Google Patents. CN Patent App. CN02, 160, 272

[14] 王石，王卫民，符建辉. 一种汉字字形认知相似度计算方法[M]. Google Patents, 28. CN Patent App. CN 201, 110, 205, 807

[15] Wang Si-li, Zhang Hua-ping, Wang Bin. Research of Optimization on Double-Array Trie and its Application[J]. Journal of Chinese Information Processing, 2006, 20: 24-30(in Chinese)
王思力，张华平，王斌. 双数组 Trie 树算法优化及其应用研究[J]. 中文信息学报，2006，20：24-30

(上接第 195 页)

[2] Liu Rui. Design of Imaging Sonar Data Acquisition and Storage System Based on NAND Flash[D]. Harbin: Harbin Engineering University, 2012(in Chinese)
柳睿. 基于 Nand Flash 的图像声纳数据采集存储系统设计[D]. 哈尔滨：哈尔滨工程大学，2012

[3] Shu Wen-li, Wu Yun-feng, Zhao Qi-yi, et al. Bad Block Management Method of NAND FLASH Memory[J]. Chinese Journal of Election Devices, 2011, 5: 580-583(in Chinese)
舒文丽，吴云峰，赵启义，等. NAND Flash 存储的坏块管理方法[J]. 电子器件，2011，5：580-583

[4] Zhang Sheng-yong, Gao Shi-jie, Wu Zhi-yong, et al. Bad Block Handle Methed of NAND Flash Memory based on FPGA[J]. Computer Engineering, 2010, 6: 239-240, 243(in Chinese)
张胜勇，高世杰，吴志勇，等. 基于 FPGA 的 NAND Flash 坏块处理方法[J]. 计算机工程，2010，6：239-240，243

[5] Li You-meng, Li Qing-cheng, Gong Xiao-li. Research and implementation of NAND-FLASH bad block management algorithm for FTL layer [A]// Proceedings of 2010 International Conference on Services Science, Management and Engineering(Volume 2)[C]. Civil Aviation University of China, 2010

[6] Han Yong-hao, Wang Shao-yun. Design and implementation of a NAND Flash dynamic bad block management algorithm [J]. Informatization Research, 2011, 37(3): 23-26(in Chinese)
韩勇豪，王少云. 一种 NAND Flash 动态坏块管理算法的设计与实现[J]. 信息化研究，2011，37(3)：23-26

[7] Lara D, Eitan Y, Ryan G. Graded Bit-Error-Correcting Codes With Applications to Flash Memory[J]. IEEE Transactions on Information Theory, 2012, 4: 2315-2327

[8] Hu Ning, Yang Qing, Wang Dong. NandFlash Memory Management Based on Blocks Set[J]. Microelectronics and Computer, 2015, 32(3): 19-22(in Chinese)
胡宁，杨琼，王冬. 基于分组的 NandFlash 块管理方法[J]. 微电子学与计算机，2015，32(3)：19-22

[9] Zhao Zhi-cao. Reliability Analysis and Optimization Design of Load-sharing Redundant System[D]. Xi'an Northwestern Polytechnical University, 2015(in Chinese)
赵志草. 共载冗余系统可靠性分析与优化设计[D]. 西安：西北工业大学，2015

[10] Gregory F. Lawler，随机过程导论(第 2 版)[M]. 张景肖，译. 北京：机械工业出版社，2010：48-61

[11] 邹逢兴，张相平，龙志强，等. 计算机应用系统的故障诊断与可靠性技术基础[M]. 中国水利水电出版社，2011：342-348